A TRANSACTIONAL RECOVERY SYSTEM

The invention relates to the field of transactional communications. Asynchronous transfer mode communication systems are being used more and more. Reliable communication is necessary in a great many cases. Reliability can be assured by adding transactional mechanisms to the communication service.

An asynchronous communication service includes a manager and one or more communication channels.

A supplier (client of the service) sends information by addressing a specific channel. The channel transfers the information to the client via its interface. The channel acts as an intermediary between two consumers and decouples information transmission at both ends. It must be possible to deliver information reliably while maintaining desynchronization of suppliers and consumers.

A transaction must satisfy four properties grouped together under the acronym ACID signifying: Atomicity, Consistency, Isolation and Durability. The four properties are closely interrelated. The objective of restart and competition control mechanisms used by a transactional engine is to ensure conformance to them.

Atomicity guarantees that either all the updates of a transaction are effected or none of them. Non-conformance to this property can change the set of data from the initial consistent state to an inconsistent state.

In order to conform to this property, any series of actions constituting a transaction is marked by a start and an end. The start of a transaction signals the start event of a transaction to the transactional engine. Two instructions are provided for marking the end of a transaction: "Commit" enables the transaction to signal to the transactional monitor that from the isolated point of the transaction all its actions have been completed successfully and "Abort" enables the transaction to signal to the transactional monitor that one or more of

its actions have failed and that it does not wish to be validated (the data modified by the transaction must be returned to its preceding state). Depending on the end of transaction event received by the transactional monitor and on its overview of the system (interaction between transactions), the transactional monitor decides whether or not to validate the transaction, i.e. whether or not to make the modifications effected by it permanent. If a system fault occurs before the end of a transaction, then the transaction is considered to be aborted. One mechanism for conforming to the atomicity property entails retaining the preceding image of any data updated for each transaction in progress. If a transaction is aborted, it is possible to undo the transaction by applying all the preceding images of the transaction.

The consistency property concerns the semantic consistency of a set of data. It can be maintained partly by mechanisms assuring integrity constraint control and by maintaining the property of the transactions. Isolation is indispensable in a multitasking environment, to guarantee that each transaction sees a consistent state of the set of data. Isolation entails guaranteeing that, if the transaction executes in parallel with other transactions accessing a common set of data, then serial execution of the same transactions would produce the same changes on the set of data accessed by the transactions. In this case, the isolation property is satisfied for that set of transactions. Durability guarantees that the updates of a validated transaction are permanent. The only action that should enable the updates of a validated transaction to be undone is executing a compensation transaction. This property goes hand in hand with the atomicity property stipulating that the updates of a transaction form a consistent whole which is either aborted entirely or validated permanently. One way to deal with a memory

or hard disc failure leading to loss of information from the database is to provide a restart mechanism to recover the lost information.

INS A2 The invention aims to improve the devices associated with durability: simple and reliable recovery systems are needed.

The use of a database is one way to obtain reliable ("crash resistant") data storage under the control of a transactional monitor. To enable asynchronous delivery of data under any circumstances, a communication service must store the data reliably between two communication steps (receiving data and sending data). The storage must offer good data writing performance.

Existing solutions use a database supplying an interface of the above kind, for example an RO- or XA-compatible interface. The data is logged in the context of a transaction. The XO and RA interfaces are examples of interfaces which enable an external transactional monitor to control the data. In the event of a data restart, this type of interface correlates the references of the external transactions with the data stored in the database in ways that are related to the transactions, so that modifications to its data can be confirmed or canceled.

Databases offer this type of interface and the associated functions but do not perform well in this particular application. They offer many other functions that are of no utility for data recovery.

INS A3 The invention provides a communication channel connecting a set of transaction initiators and consumers, associated with a logging service having an XA/RO interface to enable a transactional system to perform transactional operations on data stored in said logging service. The channels can be integrated into any transactional communication system.

The invention also provides a method of communication between a set of transaction initiators and

consumers, including intermediate communication channels. Each operation validated in a logging service having an XA/RO interface is stored to enable recovery of validated transactional operations.

Instead of a database having a transactional interface and associated with a communication channel, the invention uses a logging system offering an interface of this kind. The interface enables a conventional transactional monitor to apply correct termination of the transaction ("commit") or incorrect termination of the transaction ("rollback") to the logged objects, respectively to confirm or cancel the changes.

A logging system stores all the successive values of a variable, retaining the old values. In the event of loss of data, an older value of the data can always be recovered.

An RO or XA interface is added to enable an external transactional monitor to confirm or cancel modifications to logged data and to recover data. The logging service provides reliable storage means which can be written quickly. The added interface automates the termination of modification of data in accordance with indications supplied by an external transaction monitor.

The invention enables interworking with external transactional monitors in the event of a problem.

This additional layer in the connection service provides an XA interface for associating the transactional identifiers with the actions effected in the logging service.